Advances in Complex Systems
(2021) 2150004 (29 pages)

DOI: 10.1142/S0219525921500041

DESIGN OF CONTROLLABLE LEADER–FOLLOWER NETWORKS VIA MEMETIC ALGORITHMS

SHAOPING XIAO

Department of Mechanical Engineering,
University of Iowa,
Iowa Technology Institute (Formerly CCAD),
Iowa City, IA 52242, USA
shaoping-xiao@uiowa.edu

BAIKE SHE

Elmore Family School of Electrical and Computer Engineering,
Purdue University, West Lafayette, IN 47906, USA
bshe@purdue.edu

SIDDHARTHA MEHTA

Department of Mechanical and Aerospace Engineering,
University of Florida Research
and Engineering Education Facility,
Shalimar, FL 32579, USA
siddhart@ufl.edu

ZHEN KAN*

Department of Automation,
University of Science and Technology of China,
Hefei, Anhui 230026, P. R. China
zkan@ustc.edu.cn

Received 9 March 2021
Revised 5 August 2021
Accepted 24 August 2021
Published 25 September 2021

In many engineered and natural networked systems, there has been great interest in leader selection and/or edge assignment during the optimal design of controllable networks. In this paper, we present our pioneering work in leader–follower network design via memetic algorithms, which focuses on minimizing the number of leaders or the amount of control energy while ensuring network controllability. We consider three problems in this paper: (1) selecting the minimum number of leaders in a pre-defined network with guaranteed network controllability; (2) selecting the leaders in a pre-defined network with the minimum control energy; and

* Corresponding author.

(3) assigning edges (interactions) between nodes to form a controllable leader–follower network with the minimum control energy. The proposed framework can be applied in designing signed, unsigned, directed, or undirected networks. It should be noted that this work is the first to apply memetic algorithms in the design of controllable networks. We chose memetic algorithms because they have been shown to be more efficient and more effective than the standard genetic algorithms in solving some optimization problems. Our simulation results provide an additional demonstration of their efficiency and effectiveness.

Keywords: Leader–follower networks; memetic algorithms; controllability; control energy.

1. Introduction

Complex systems [44], which are composed of many components or agents interacting with each other, are omnipresent in nature and man-made systems. The behavior of a complex system is intrinsically difficult to model because of the various types of interactions or communications between its components or between the system and its environment. One useful approach is to represent a complex system as a network [13, 32], in which the nodes represent the components (i.e. the agents) while the links (i.e. the edges) represent the interactions/communications between the components. There are two types of interactions in complex networks: cooperative and antagonistic (i.e. competitive). Unsigned graphs with positive edge weights are normally used to represent networks with cooperative interactions only. Such graphs can be either directed or undirected to imitate one-way or two-way communications between agents. On the other hand, signed graphs allow antagonistic interactions with negative edge weights in addition to cooperative interactions. Such networks are also called coopetition networks because both cooperative and competitive interactions exist in complex systems such as multi-agent systems [62].

Many efforts have been devoted to studying the structural and functional properties of complex networks with a leader–follower framework [15]. In this framework, the leader nodes dictate the overall behavior of the network by influencing or controlling the follower nodes via the edge (i.e. interaction) characteristics of the network. The controllability of complex networks [3, 4, 31] has been extensively studied in the literature. Tanner [60] presented one of the first works to discuss the structural controllability [14] of leader–follower networks. Graph theory [2, 22, 50, 53, 56] has become an approach to studying network controllability that can be characterized based on the spectral analysis of the system matrix. In addition, necessary and sufficient conditions for network controllability have been developed for tree topology [17, 23, 24, 51], grid graphs [41], and path/cycle graphs [45].

Various metrics of network controllability [46] have been developed to characterize the control energy of complex networks. One of the most used metrics is the controllability Gramian [9, 64], which is related to the system matrices of complex networks. The properties of the controllability Gramian, including the minimum eigenvalue, the trace of its inverse, and the condition number, have been employed to study the energy-related performance in network control [55, 58, 68]. Olshevsky [43] proved that the clustered eigenvalues of a system matrix could result in the need for

significant control energy. Zhao and Pasqualetti [69] determined the necessary and sufficient graphical conditions for a discrete-time dynamical network to feature a diagonal Gramian so that the network topology and weights could be identified.

The design of leader–follower networks, especially leader selection, has been of great interest. Greedy heuristic search methods [42, 47, 63] have commonly been used to select leaders to ensure the controllability of given networks, and optimization-based leader selection for the minimum control energy [10, 48] has been presented as well. Aguilar and Gharesifard [1] introduced a graph-theoretic classification for leader–follower control systems and studied cases in which the systems are uncontrollable after selecting leader nodes. She *et al.* [54] developed graph-inspired characterizations of energy-related controllability based on the network topology and the agent dynamics. Also, Becker *et al.* [6] addressed the constrained design of linear systems to improve system controllability and energy efficiency.

Genetic algorithms (GAs), i.e. a subclass of evolutionary algorithms (EAs) [20], apply the principles of evolution found in nature to the problem of finding an optimal solution. They have been widely used in engineering problem-solving, including multi-agent systems [7, 34, 40], optimal control problems [38], and optimal sensor/actuator locations. It is worth mentioning that Li and Lu [29] conducted the first work to employ a GA to optimize the controllability of arbitrary networks. Their problem was formulated to choose the minimal control nodes (leaders) such that the network can be fully controlled.

Unlike some other heuristic search methods in which one guessed solution is used to start the optimization process, EAs start with a population of randomly sampled solutions. In GAs, solutions to a given problem are defined by so-called chromosomes, which are often represented by bit strings, especially binary strings. Each solution is associated with a so-called fitness that can be calculated according to the objective or cost function. The evolution of chromosomes due to the actions of reproduction, mutation, and natural selection are iterated until one or more optimal solutions with the highest fitness value are found. It has been found that nave EAs are not well suited to fine-tuned searches in complex combinatorial spaces [11, 12] due to premature convergence to local optima and that hybridizing EAs with other techniques [8, 16, 49] can significantly improve their performance.

Memetic algorithms (MAs) [27], which are a combination of local search [33] with EAs, have received a lot of attention from researchers because they can prevent the loss of population diversity. Merz and Freisleben [37] demonstrated that MAs are both more efficient (i.e. achieving fast convergence) and more effective (i.e. finding global optima) than standard GAs and other heuristic search methods. Wrona and Pawełczyk [65] adopted a controllability-oriented approach to place the actuators on a fully clamped isotropic rectangular plate and employed an MA to find efficient locations for actuators. In another study, Tang and co-workers [59] utilized an MA to design optimal networks that would be resistant to both targeted and random attacks.

Many works have discussed the convergence of EAs from different perspectives [18, 52]. Particularly, Leung *et al.* [28] proposed a so-called abstract EA (AEA) to unify various EAs, including GAs. In their work, the evolution was described via AEA as an abstract stochastic process composed of selection and evolution operators. After axiomatically characterizing the operators properties, they established several universal convergence theorems for EAs. To generalize these results from EAs to MAs, Xu and He [67] modeled the local search in MAs as a strong evolution operator and extended the AEA model to the case of MAs. As a result, an abstract model and the corresponding convergence theorems for MAs were established [67]. Another research [37], including the work of Krasnogor and Smith [27], demonstrated that MAs had faster convergence rates than GAs.

In this work, based on the MA framework, a novel leader selection and edge assignment approach is developed for the optimal design of controllable leader–follower networks. Specifically, this work addresses the problems of (1) selecting the minimum number of leaders in a pre-defined network with guaranteed network controllability; (2) selecting the leaders in a pre-defined network with the minimum control energy; and (3) assigning edges (interactions) between nodes to form a controllable leader–follower network with the minimum control energy.

This work makes three major contributions as follows:

- It is the first to apply MA in leader–follower network design. Li and Lu [29] pointed out that the standard GA tended to converge toward local optima when selecting leader nodes to design networks. MA, on the other hand, can maintain population diversity while achieving fast convergence.
- The proposed MA framework can be applied to design arbitrary networks, including signed, unsigned, directed, or undirected networks. In contrast to an earlier study [65], which applied MA in a specific problem (i.e. finding optimal locations of actuators on a rectangular vibrating plate), our work considers general network design problems via either leader selection or edge assignment while ensuring network controllability and the minimum control energy. Additionally, our work targets the design of leader–follower networks. This differs from another earlier work [59] that optimized small-world and scale-free networks.
- The proposed MA framework has great potential to conclude all optimal designs in one simulation. In our formulated problems, it is common that the optimal design is not unique. Our simulations demonstrate that most GA simulations cannot find all optimal designs individually. In other words, multiple GA simulations are needed to collect all designs. In contrast, MA has a very high chance of finding all optimal designs in a single simulation.

This paper is organized as follows. Section 2 describes the Laplacian dynamics of leader–follower networks, as well as the network controllability and control energy. Section 3 formulates the network design problems and describes the MA framework

in leader–follower network design. A few simulations with discussions are presented in Sec. 4, followed by conclusions and future work in Sec. 5.

2. Leader–Follower Networks and Controllability

2.1. *Leader–follower networks*

Consider a networked system captured by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ with the node set $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ and the edge set $\mathcal{E} = \{e_{ij} | 1 \leq i, j \leq n\} \subseteq \mathcal{V} \times \mathcal{V}$. The interactions within the network are captured by the adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$ with $a_{ij} = 1$, $a_{ij} = -1$, $a_{ij} = 0$ representing the cooperative, competitive, noninteractions, respectively, from node v_j to node v_i. No self-loop is considered, i.e. $a_{ij} = 0$ $(i = 1, \ldots, n)$. A graph is deemed unsigned if the adjacency matrix contains only nonnegative entries. Otherwise, it is defined as a signed graph. Both directed and undirected edges are allowed in the graph. A directed edge captured by a_{ij} indicates that the information is passed from node v_j to node v_i only. An edge is defined as undirected if nodes v_i and v_j can exchange information with each other in the same way, represented by $a_{ij} = a_{ji}$. An undirected graph contains only undirected edges, while a directed graph in this work may have both directed and undirected edges. The neighbor set of v_i is defined as $\mathcal{N}_i = \{v_j | (v_i, v_j) \in \mathcal{E}\}$, which contains the nodes that receive the information from node v_i. The in-degree of node v_i is defined as $d_i = \sum_j |a_{ij}|$ to represent the number of nodes from which node v_i receives the information. The Laplacian matrix of a graph $\mathcal{G}$ is defined as $\mathcal{L}(\mathcal{G}) = \mathcal{D} - \mathcal{A}$, where $\mathcal{D}$ is a diagonal matrix composed of the in-degree of the nodes, defined as $\mathcal{D} = \text{diag}\{d_1, d_2, \ldots, d_n\}$.

A typical leader–follower networked system can be formed by dividing the node set into a leader set $\mathcal{V}_l \subset \mathcal{V}$ and a follower set $\mathcal{V}_f \subset \mathcal{V}$ such that $\mathcal{V}_l \cup \mathcal{V}_f = \mathcal{V}$ and $\mathcal{V}_l \cap \mathcal{V}_f = \emptyset$. Figure 1 shows leader–follower networked systems captured by three types of networks: an unsigned directed network A, an unsigned undirected network

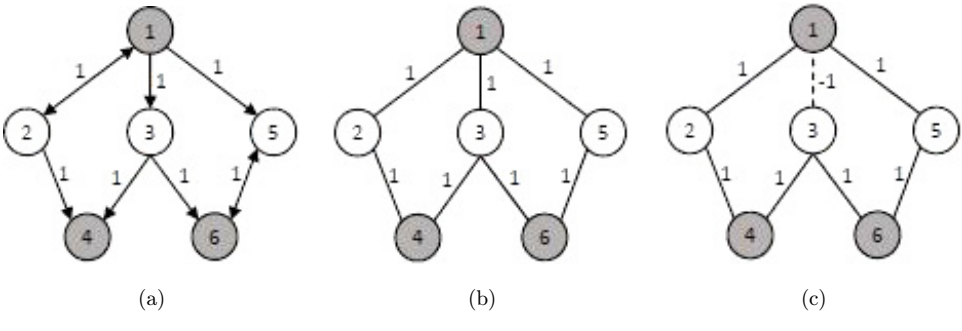


Fig. 1. Three leader–follower networks: (a) unsigned directed network A; (b) unsigned undirected network B; and (c) signed undirected network C.

B, and a signed undirected network C. Networks B and C have the same graph topology. However, the cooperative relationship between nodes 1 and 3 in network B is changed to a competitive relationship in network C, represented by the dashed line with a unit negative weight.

2.2. *Laplacian dynamics*

Let the state of each node in a network $\mathcal{G}$ be denoted by a state vector $\boldsymbol{x} = [x_1\ x_2\ \ldots\ x_n]^T$, where x_i represents the state of node v_i and the nodes interact with each other through linear consensus rules. Hence, the system dynamics is given by the following equation [39]:

$$\dot{\boldsymbol{x}} = -\mathcal{L}\boldsymbol{x}, \tag{1}$$

where $\mathcal{L} = \mathcal{L}(\mathcal{G})$ is called the Laplacian matrix of $\mathcal{G}$. The Laplacian dynamics indicates that each node updates its state through the states of its neighboring nodes.

2.3. *Network controllability*

Consider a leader–follower networked system captured by (1), and let N_f and N_l denote the number of followers and the number of leaders, where $N_f + N_l = n$. If the first N_f nodes of the system are organized as followers with the remaining N_l nodes being leaders, the Laplacian dynamics in (1) can be represented as the dynamics of the leaders and followers separately

$$\begin{bmatrix} \dot{\boldsymbol{x}}_f \\ \dot{\boldsymbol{x}}_l \end{bmatrix} = -\begin{bmatrix} \mathcal{L}_f & \mathcal{L}_{fl} \\ \mathcal{L}_{lf} & \mathcal{L}_l \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_f \\ \boldsymbol{x}_l \end{bmatrix}, \tag{2}$$

where $\boldsymbol{x}_f \in \mathbb{R}^{N_f \times 1}$ and $\boldsymbol{x}_l \in \mathbb{R}^{N_l \times 1}$ denote the states of followers and leaders, respectively. In addition, $\mathcal{L}_f \in \mathbb{R}^{N_f \times N_f}$, $\mathcal{L}_{fl} \in \mathbb{R}^{N_f \times N_l}$, $\mathcal{L}_{lf} \in \mathbb{R}^{N_l \times N_f}$, and $\mathcal{L}_l \in \mathbb{R}^{N_l \times N_l}$ represent the follower-to-follower, leader-to-follower, follower-to-leader, and leader-to-leader matrices. Because the states of the leaders are determined directly by external control inputs, the dynamics of the followers is given by

$$\dot{\boldsymbol{x}}_f = -\mathcal{L}_f\boldsymbol{x}_f - \mathcal{L}_{fl}\boldsymbol{x}_l. \tag{3}$$

The network connectivity is assumed to be unchangeable. Therefore, (3) represents a linear time-invariant system, and the controllability matrix of (3) can be expressed as [19]

$$\mathcal{C} = [-\mathcal{L}_{fl} \quad \mathcal{L}_f\mathcal{L}_{fl} \quad \ldots \quad (-1)^{N_f}\mathcal{L}_f^{N_f-1}\mathcal{L}_{fl}]. \tag{4}$$

According to the Popov–Belevitch–Hautus rank condition or the Kalman rank condition, if the controllability matrix in (4) has a full row rank, i.e. the rank equals the number of followers N_f, the leader–follower networked system in (3) is

controllable, which is referred to as network controllability in this work. For example, in Fig. 1, network A is not controllable, while networks B and C are controllable. As shown in (3), the controllability greatly depends on the structure of the leader–follower system, i.e. the graph topology and the locations of the leaders.

2.4. *Control energy*

The rank of the controllability matrix in (4) can be used to determine whether a network is controllable with appropriately selected leaders and/or assigned edges (i.e. interactions). However, for a leader–follower networked system, controllability can be ensured with different groups of leaders. This raises the question of how difficult it is to control the network or how much energy is needed to drive the network to the goal state with different groups of leaders. Since the state of the leaders is determined directly by external control inputs, through taking $\boldsymbol{x}_l$ as the exogenous control signal, the total control energy over the time interval $[0, t]$ is defined as follows to provide an energy-related quantification of network control:

$$E(t) = \int_0^t \|\boldsymbol{x}_l(\tau)\|^2 d\tau, \tag{5}$$

where $\|\boldsymbol{x}_l(\tau)\|$ represents the Euclidean norm of $\boldsymbol{x}_l$. Assuming the initial state $\boldsymbol{x}_f(0) = 0$ and the optimal control $\boldsymbol{x}_l$ [25], the minimum control energy required to drive the system in (3) to the desired goal state $\overline{\boldsymbol{x}}_f$ can be calculated as

$$E(t) = \overline{\boldsymbol{x}}_f^T \mathcal{W}_K^{-1}(t) \overline{\boldsymbol{x}}_f, \tag{6}$$

where $\mathcal{W}_K$ is the controllability Gramian at time t and can be computed as

$$\mathcal{W}_K(t) = \int_0^t e^{-\mathcal{L}_f \tau} \mathcal{L}_{fl} \mathcal{L}_{fl}^T e^{-\mathcal{L}_f^T \tau} d\tau. \tag{7}$$

The controllability Gramian $\mathcal{W}_K(t)$ is positive definite if and only if the system in (3) is controllable [19]. The infinite-horizon Gramian $\mathcal{W}_K = \mathcal{W}_K(t \to \infty)$ is adopted here due to the asymptotic or exponential stability of the dynamic systems considered in the work. The controllability Gramian $\mathcal{W}_K(t)$ provides an energy-related measure of network controllability, and various quantitative metrics have been developed [58]. One of them is the trace of the inverse controllability Gramian, i.e. $\text{trace}(\mathcal{W}_K^{-1})$, which is proportional to the average control energy required to move the system state to the desired state. In this paper, we use $\text{trace}(\mathcal{W}_K^{-1})$ to represent the average control energy in the studied networks. Consider the controllable networks B and C in Fig. 1 as examples: although they have the same controllability, the different interactions between nodes 1 and 3 in the two networks result in different control energy. The average control energy in network B is 26.73, while it is 7.33 in network C.

3. Methodology

3.1. *Problem formulation*

Given a network $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{A})$ in which the network connectivity $\mathcal{E}$ and the adjacency matrix $\mathcal{A}$ are prescribed, the objective is selecting the leaders or assigning the edges while ensuring network controllability. The problem can be formulated as

$$\min(\Phi) \quad \text{subject to} \quad \text{rank}(\mathcal{C}) = N_f, \tag{8}$$

where Φ is the objective function, $\mathcal{C}$ is the controllability matrix defined in (4), and N_f is the number of followers.

We consider three optimization problems when designing controllable leader–follower networks:

(i) Selecting the minimum number of leaders: $\Phi = N_f$.
(ii) Selecting the leaders with the required number of leaders N_q to achieve the minimum control energy: $\Phi = \text{trace}(\mathcal{W}_K^{-1})$ where $\mathcal{W}_K$ is the controllability Gramian defined in (7).
(iii) Assigning the edges to achieve the minimum control energy.

3.2. *Memetic algorithms*

MAs are extensions of EAs that apply local searches on each generated population to refine individual solutions by improving their fitness values. They are also population-based approaches, and Fig. 2 illustrates a general flowchart of MAs. An MA simulation starts with a population of randomly generated candidate solutions, which are also called individuals. At each iteration, some individuals from the existing population are selected as parents. The selection depends on the fitness values evaluated for each individual. The individual with a higher fitness value is more likely to be selected. Then, three operations — reproduction, mutation, and local search — are applied to generate a new population in which the individuals are called children or offspring. A crossover technique commonly used in reproduction is adopted in this paper to exchange some genes of parent chromosomes to create new chromosomes for children. Then, the mutation operation is conducted according to a prescribed probability, which is often very low, to maintain genetic diversity. Finally, a local search technique, e.g. hill-climbing, is utilized to improve the fitness values of selected children. Only a portion of the generated individuals is selected to maintain

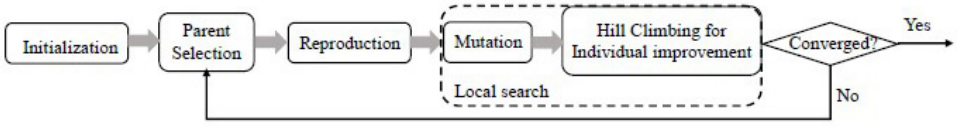


Fig. 2. MA flowchart.

the balance between exploration (the degree of evolution) and exploitation (individual improvement) [21]. After the new population is generated, the above procedure will be repeated until a termination criterion is met. Individuals with the highest fitness value in the last population are the final solutions.

Using local search techniques can allow MAs to avoid premature convergence to suboptimal solutions or locally optimal solutions, which commonly occurs in nave EAs. Some research has been carried out in performance analyses of MAs [36, 61] by solving various optimization problems. It has been shown that MAs are more efficient (achieving fast convergence) and more effective (finding global optima) than EAs and some other heuristic search methods [5, 37]. In addition, multiple optimal solutions may exist in some optimization problems, including the network design problems discussed in this paper. Our simulations in the next section demonstrate that MAs have a much higher possibility of finding all of the optimal solutions in one simulation (i.e. one run) than GAs because MAs can greatly maintain the population diversity via the local search operation.

3.3. *Memetic algorithms for leader–follower network design*

The fitness function depends on the problem objective and plays a significant role in MAs. The higher the fitness value a solution has, the larger the chance it will be selected as a parent for reproduction. Consequently, the individuals in the current population with higher fitness values have higher probabilities of passing their genes to the next population, as in the principles of evolution found in nature. Various fitness functions are designed for the different optimization problems defined in Sec. 3.1, as follows.

(i) Selecting the minimum number of leaders

$$F_1(s) = \frac{1}{N_l + \alpha_1[\text{rank}(\mathcal{C}) - N_f]^2}, \tag{9}$$

where s is a candidate solution and α_1 is the penalty parameter to enforce the controllability constraint, $\text{rank}(\mathcal{C}) = N_f$. The penalty parameter can be $10N_l$ as used by Li and Lu [29] or any large number, e.g. $\alpha_1 = 1000$ in this paper.

(ii) Selecting the leaders with the required number of leaders N_q to achieve the minimum control energy

$$F_2(s) = \frac{1}{\text{trace}(\mathcal{W}_K^{-1}) + \alpha_2[\text{rank}(\mathcal{C}) - N_f]^2 + \beta_2[N_l - N_q]^2}, \tag{10}$$

where α_2 is the penalty parameter to enforce the controllability constraint while β_2 is the parameter to enforce the required number of leaders, i.e. $N_l = N_q$. To achieve the objective of minimizing the control energy, it is necessary to choose a number much larger than the average control energy, e.g. $\alpha_2 = \beta_2 = 10\text{trace}(\mathcal{W}_K^{-1})$, as the penalty parameters, which can be updated during the optimization.

(iii) Assigning the edges to achieve the minimum control energy

$$F_3(s) = \frac{1}{\text{trace}(\mathcal{W}_K^{-1}) + \alpha_3[\text{rank}(\mathcal{C}) - N_f]^2}, \tag{11}$$

where α_3 is defined as the same as α_2 in (10).

One thing common to network design via leader selection is that the solutions select some nodes in $\mathcal{V}$ as the leaders and others as the followers while satisfying the problem objectives described in (8). To apply MAs, a candidate solution is represented by an n-bit binary string as its chromosome, in which each bit represents the node status, i.e. 1 for leading and 0 for following. For example, all networks in Fig. 1 have the same node set $\mathcal{V}$ with six nodes, in which nodes 1, 4, and 6 are the leaders. Therefore, they have the same chromosome {1 0 0 1 0 1}.

The operations of reproduction and mutation in MAs are illustrated in Fig. 3. The crossover technique is applied to two selected parent solutions to reproduce two child solutions. Using a six-node network as an example, two selected parents have different chromosomes, {1 0 0 1 0 0} and {0 1 1 0 1 1}. The crossover point is randomly generated at the fourth bit. The parents switch the genes up to the crossover point as shown in Fig. 3(a). Consequently, two child solutions are generated via this reproduction operation as {0 1 1 0 0 0} and {1 0 0 1 1 1}. In addition, each generated child solution has a very low probability (usually 1–5%) to be mutated. In the mutation operation, shown in Fig. 3(b), a randomly selected bit (gene) is switched to the opposite status, i.e. from 0 to 1 or from 1 to 0.

Stochastic hill-climbing techniques [66] can be used in the local search operation to improve an individual's fitness. A simple hill-climbing, i.e. bit-climbing, as an extension of the mutation operation is employed in this paper. After a new population is generated, each individual has a moderate probability to participate in a

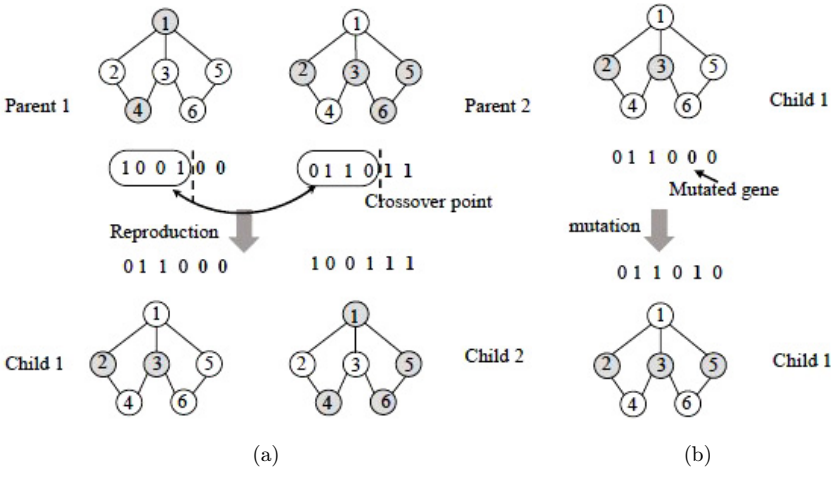


Fig. 3. (a) Reproduction and (b) mutation in MAs.

local search in which a randomly selected bit (gene) switches status. The newly searched solution will replace the original one only if the new one has a higher fitness value. Since the network connectivity is pre-defined in the above problems, the proposed MA framework can be applied to both directed and undirected networks.

To design a network via edge assignments, the chromosomes employed in MAs are different for directed or undirected networks even though they have the same network topology. The chromosome of a candidate solution for a directed network can be directly generated from the network connectivity $\mathcal{E}$. Taking network A in Fig. 1 as an example, where the edges represent possible interactions, the edge set is $\mathcal{E}_A = \{e_{12}, e_{21}, e_{31}, e_{42}, e_{43}, e_{51}, e_{56}, e_{63}, e_{65}\}$. The corresponding chromosome is a 9-bit binary string $\{a_{12}\ a_{21}\ a_{31}\ a_{42}\ a_{43}\ a_{51}\ a_{56}\ a_{63}\ a_{65}\}$ in which $a_{ij} = 1$ to assign the communication from node v_j to node v_i and $a_{ij} = 0$ to prohibit communication. It shall be noted that a_{ij} is also an entry of the adjacency matrix $\mathcal{A}$ as defined in Sec. 2.1. Therefore, the chromosome determines the adjacency matrix $\mathcal{A}$ and then the Laplacian matrix $\mathcal{L}(\mathcal{G})$ to evaluate the network controllability and the control energy for fitness calculation in (11).

In an undirected network, e_{ij} in the edge set represents the potential communication between nodes v_i and v_j. In the example of network B in Fig. 1, the edge set is $\mathcal{E}_B = \{e_{12}, e_{13}, e_{15}, e_{24}, e_{34}, e_{36}, e_{56}\}$ and the chromosome is a 7-bit binary string, $\{c_{12}\ c_{13}\ c_{15}\ c_{24}\ c_{34}\ c_{36}\ c_{56}\}$ in which $c_{ij} = 1$ or 0 indicates the presence or absence of interaction between nodes v_i and v_j. In this case, the entries of the adjacency matrix $\mathcal{A}$ can be determined via $a_{ij} = a_{ji} = c_{ij}$.

3.4. *Complexity analysis*

The computing complexity of population-based evolutionary approaches, including GAs and MAs, largely depends on the fitness function. Given an MA simulation with the number of generations G, the size of population P, and the length of chromosome L, the space complexity is $O(PL + O(\text{fitness}))$ where L can be the number of nodes (n) or the number of edges (n_e) for the problems of selecting leaders or assigning edges, respectively. The fitness functions in (9)–(11) include the controllability matrix $\mathcal{C}$ and/or controllability Gramian $\mathcal{W}_K$ so its space complexity is $O(\text{fitness}) = O(N_f^2 N_l + N_f^2) = O(N_f^2 N_l)$, in which N_f is the number of followers and N_l is the number of leaders. Also, considering fitness evaluation, one-point crossover, point mutation, bit-climbing local search, and a stochastic selection for sorting the population used at each generation in our approach, the time complexity of MAs is $O(G(PO(\text{fitness}) + PL + PP_m + PP_l O(\text{fitness}) + P\log P)) = O(GPO(\text{fitness}) + GP\log P)$ where P_m is the mutation probability and P_l is the local search probability. The computing of fitness includes generating the controllability matrix in (4) $(O(N_f^3 N_l))$, determining the rank of the controllability matrix via Gaussian elimination $(O(N_f^3 N_l))$, and calculating the controllability Gramian $(O(N_f^3))$ [46] in (7). Therefore, the time complexity of fitness is in the order of $O(N_f^3 N_l)$, and consequently, the time complexity of MAs is in the order of $O(GPN_f^3 N_l + GP\log P)$.

4. Simulations and Results

4.1. *Selecting the minimum number of leaders*

First, we use a 15-node network to demonstrate the application of MAs in controllable network design and to compare the performances of the MA and the GA. The signed tree graph of this 15-node network is shown in Fig. 4; the solid line represents the cooperative interaction (with a positive unit weight), and the dashed line represents the competitive interaction (with a negative unit weight). It should be noted that the MA framework proposed in this paper can handle any signed, unsigned, directed, or undirected network. The objective in our first example is to select the minimum number of leaders while ensuring the controllability of the complex network, as formulated in (8). Any proposed solution is defined by a 15-bit binary string, $s = \{n_1, n_2, \ldots, n_{15}\}$, which forms a chromosome in the GA or MA. The digit n_i can be either 0 or 1, which represents the status of node v_i as a follower or a leader

Both GA and MA simulations are conducted in order to investigate their performances. Each simulation starts with an initial population of 100 guessed solutions. At each iteration, the individual solutions are ranked based on the fitness calculated via (9). Forty top-ranked solutions must be included in the parent group in addition to the randomly selected 60 solutions from the whole population. Then, each randomly assigned pair of parent solutions produces two child solutions via the crossover technique, and each child solution has a 2% probability of being mutated. A bit-climber local search is employed in MA (not GA) simulations so that each child solution has a chance to move to its neighboring state with a higher fitness value. The produced child solutions form a new population as the next generation.

In this example, there are four optimal solutions that have only one leader node. The optimal solutions result in the highest fitness value of 1.0. Figure 5 shows the

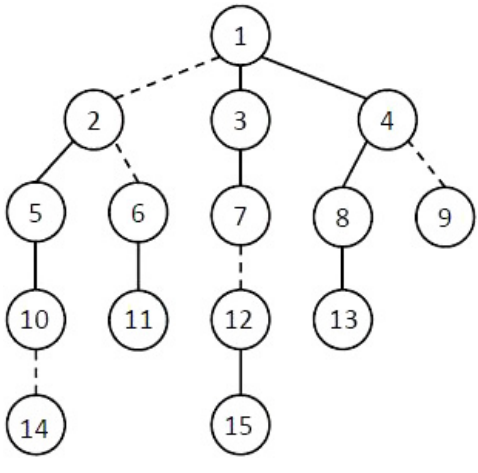


Fig. 4. The signed tree graph of a 15-node network.

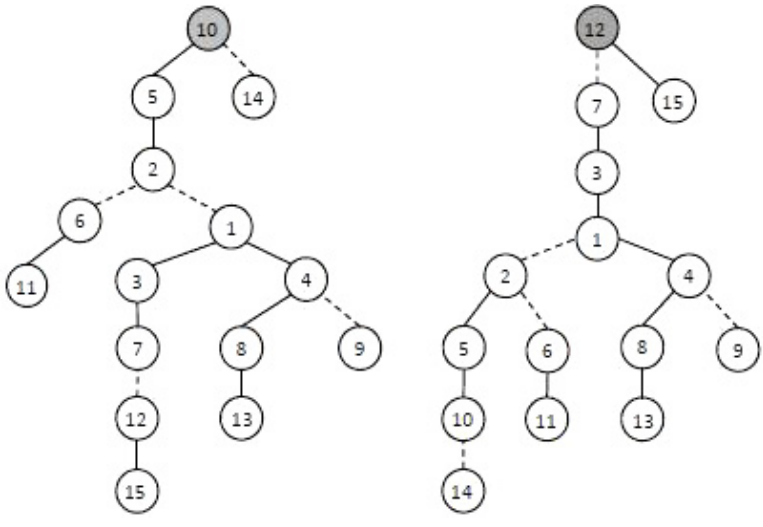


Fig. 5. Controllable 15-node networks with one selected leader.

tree graphs after selecting either node 10 or node 12 as the leader. The other two solutions include either node 14 or node 15 as the leader node. All of the selections result in controllable networks. Although GA can provide the same solutions, its two drawbacks, mentioned in Sec. 3.2, are confirmed in this example when compared to MA simulations. First, GAs converge more slowly than MAs, and second, a single GA simulation has difficulty concluding all the optimal solutions due to premature convergence.

We have conducted 120 simulations via MAs and GAs, respectively. Figure 6 illustrates the average fitness of the population at each iteration. It is evident that MA simulations converge much faster than GA simulations. This agrees with the

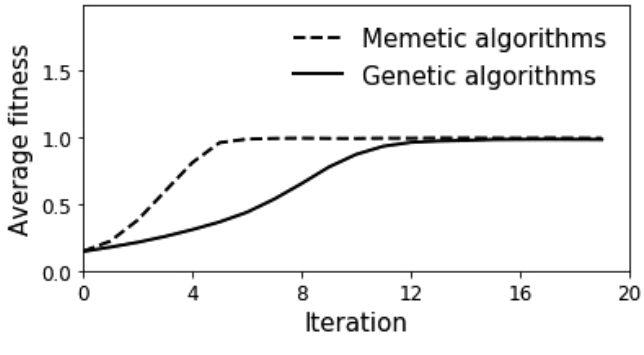


Fig. 6. Evolution of fitness in MA and GA simulations.

Table 1. The success ratios of finding optimal solutions at one simulation via GAs and MAs.

The number of optimal solutions (out of four)	GAs (%)	MAs (%)
4	24.6	88.3
3	40.0	10.8
2	30.0	0.9
1	5.4	0.0

conclusions from many other MA applications, including those of Krasnogor and Smith [27] and Merz and Freisleben [37].

On the other hand, MAs have been shown to reduce the likelihood of premature convergence to local optima [37]. Furthermore, our study finds that MAs have great potential to conclude all of the optimal solutions in a single simulation but that most GA simulations can only reach some (not all) of the optimal solutions. Table 1 compares GAs and MAs on the success ratios of finding various number of optimal solutions (up to a total of four) at one simulation. Among the 120 GA simulations in this example, only 24.6% can reach all four optimal solutions; 40% and 30.8% find three and two solutions, respectively. The other GA simulations can only find one optimal solution. In contrast, 88.3% of the 120 MA simulations can find all four optimal solutions; 10.8% find three solutions and 0.9% find two solutions. This advantage of MAs is due to the implementation of the local search on some individuals of the new generation to reach a balance between exploration and exploitation [21]. As a result, the different individual solutions may tend to converge to different optimal states to enhance population diversity. In a scenario with many optimal solutions, MAs have the potential to find all of them if the initial population is large enough.

4.2. *Selecting leaders for the minimum control energy*

In the previous example, a single properly selected leader can control the 15-node network shown in Fig. 5. However, a single leader may result in relatively high control energy. As described in Sec. 2.4, the average control energy can be calculated as $\text{trace}(\mathcal{W}_K^{-1})$, where $\mathcal{W}_K$ is the controllability Gramian matrix in (7). Table 2 lists the average control energies of the 15-node networks when a single node is selected as the leader.

Given the complex network and the required number of leaders, the objective in this example becomes assigning the leaders so that the control energy is minimized while the complex network remains controllable. The problem is formulated as (8), and the fitness of a candidate solution can be calculated via (10). Considering different numbers of leaders up to 7, we employ MAs to select leaders so that the average control energy of the network is the minimum while ensuring network controllability. The initialization and iteration procedure in MA simulations are the same as described in the previous section. Figure 7 illustrates the graphs of designed

Table 2. The average control energy of 15-node networks with a single leader node.

Leader node	Average control energy
Node 10	3.12×10^{13}
Node 12	5.94×10^{13}
Node 14	2.15×10^{14}
Node 15	5.10×10^{14}

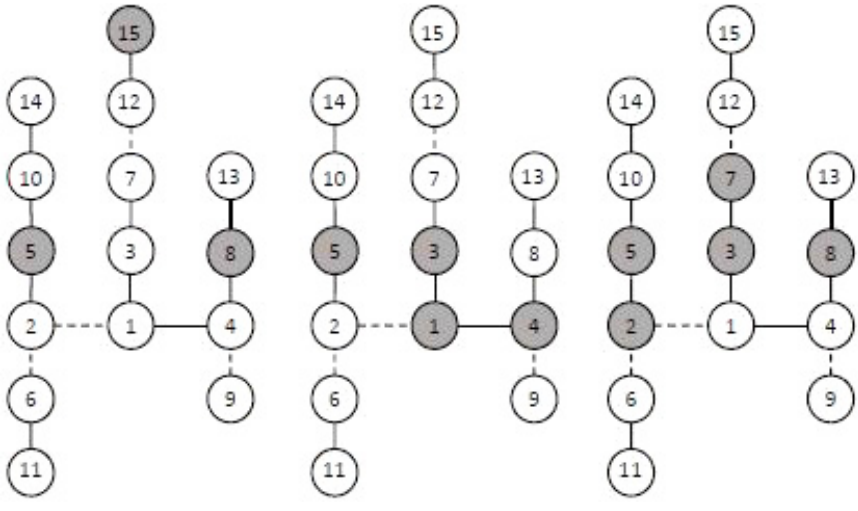


Fig. 7. The 15-node networks with 3 leaders (left), 4 leaders (middle), and 5 leaders (right).

15-node networks with 3, 4, and 5 leaders. Table 3 includes the optimal leader selections and the corresponding control energy. It can be seen that more leaders result in less control energy.

4.3. *Assigning edges*

Here, we design controllable 15-node networks when the leaders are pre-selected but the communications between nodes need to be determined. The problem becomes

Table 3. The minimum control energy of 15-node networks with different number of leaders.

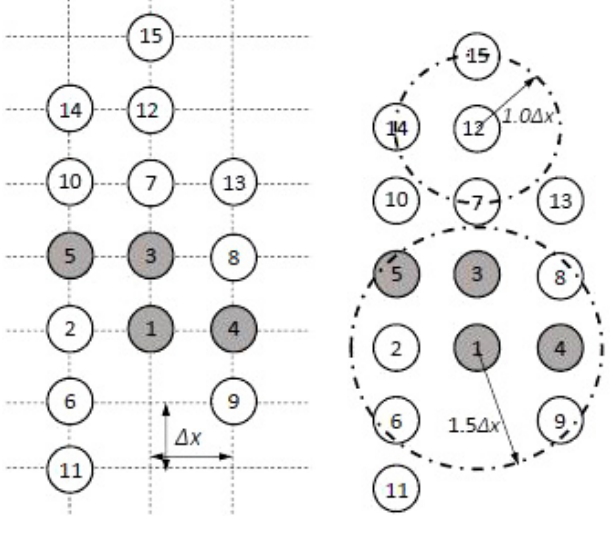
The number of leaders	Selected leaders	The minimum control energy
1	[10]	3.12×10^{13}
2	[2, 12]	7.43×10^{6}
3	[5, 8, 15]	2.45×10^{4}
4	[1, 3, 4, 5]	679.28
5	[2, 3, 5, 7, 8]	161.65
6	[1, 2, 5, 7, 8, 12]	69.35
7	[2, 4, 5, 6, 7, 12, 13]	38.99

assigning edges, i.e. communications or interactions, between nodes so that the minimum control energy is achieved. It is assumed that each node can only communicate with its neighbors within a so-called communication domain, which determines the neighboring nodes with which this node can communicate. It should be noted that the communication domain can be defined in either physical space or cyberspace (e.g. connections in social contexts) via the Euclidean distance. The ways to identify the communication domains may vary in different problem domains, and the communication domain can be changed with time, especially for a network of multiple autonomous vehicles or mobile robots. We consider only time-invariant communication domains in this paper.

The approach we employ here is to define a circular communication domain via its radius calculated as

$$r_i = \gamma \min_{i,j \neq i} d_{ij}, \tag{12}$$

where r_i is the radius of the communication domain of node v_i, d_{ij} is the Euclidean distance between node v_i and node v_j, and γ is a scalar to determine the communication domain's size. Here consider 15 nodes located at grid points of a square lattice in cyberspace as shown in Fig. 8(a). Nodes 1, 3, 4, and 5 are assigned as the leaders. The minimum distance between the two neighboring nodes is $\triangle x$. As depicted in Fig. 8(b), if $\gamma = 1.0$ (i.e. the von Neumann neighborhood) is chosen to determine the communication domain of node 12, the neighbors it might communicate with include nodes 7, 14, and 15. In another case, if $\gamma = 1.5$ (i.e. Moore neighborhood) is chosen for node 1, its neighbors include nodes 2–6, 8, and 9.



(a) (b)

Fig. 8. Fifteen nodes located in a cyberspace.

We first consider unsigned undirected networks where $\gamma = 1.0$ for all nodes and each node has up to four neighbors. There are 19 possible interactions in the 15-node network shown in Fig. 8(a). Therefore, in MAs, the network edge assignment is determined by a 19-bit binary string, $s = \{c_{12}\ c_{13}\ c_{14}\ c_{25} \ldots\}$, where c_{ij} represents the communication status (1 for communication or 0 for noncommunication) between nodes v_i and v_j. The objective is assigning the edges between nodes so that the control energy is the minimum and the complex network is controllable. The fitness can be calculated via (11).

Eight solutions are found via the MA, and Fig. 9(a) shows the parent network architecture. The others are the offspring of this "parent" network after removing one or more connections between the leaders. Two of them are shown in Fig. 9(b). The original design in Fig. 7 (middle) is one of the offspring. This agrees with the conclusions from She *et al.* [55] in which removing connections between leaders does not affect the controllability and control energy of leader–follower networks. Also, unsigned and signed networks have the same controllability and the same control energy (i.e. 679.28 in this example) if they have the same network topology.

If γ is increased to 1.5, each node will be able to communicate with up to eight neighbors, as shown in Fig. 8(b). The chromosome will be extended to a 33-bit binary string. There are more than 50 designs that can achieve the minimum control energy while ensuring the network controllability. Ideally, given a large enough initial population, MA will be able to find all of the possible solutions. A few sample designs are shown in Fig. 10. Analyzing the designs in Fig. 10, we notice that more edges are

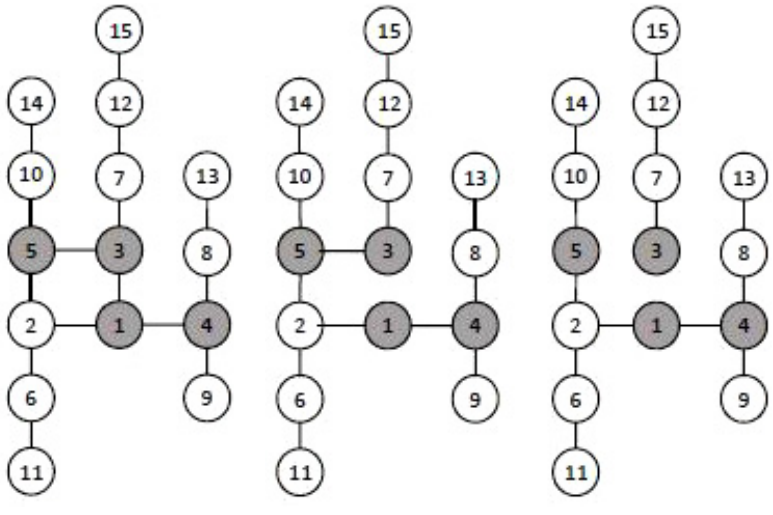


Fig. 9. Edge assignments of undirected 15-node networks with the minimum control energy ($\gamma = 1.0$).

Table 4. The number of leaders (N_l) and the number of followers (N_f) with various degrees in the designs of Fig. 10.

	Design 1		Design 2		Design 3	
Degree	N_l	N_f	N_l	N_f	N_l	N_f
5	1		2			
4	1		1		1	
3	1	1	1	1	3	1
2		4		4		4
1		6		6		6

assigned to the leaders than the followers so that the leaders become high-degree nodes. The numbers of leaders and followers with various degrees are listed in Table 4. Figure 10 also illustrates that the followers connect with at most two other followers. It shall be noted that node 7, in all three designs, connects with two leaders and only one follower although it has three edges. On the other hand, in the designs in Fig. 10 ($\gamma = 1.5$), only one follower has two edges connecting with other followers. However, the number of followers that have two follower–follower edges is two in the designs in Fig. 9 ($\gamma = 1.0$). Therefore, the designs in Fig. 10 have lower control energy (543.28) than the one (679.28) associate with the designs in Fig. 9. It should be noted that additional objectives, such as the minimum number of edges, can be employed to reduce the number of optimal designs via multi-objective optimization techniques [26].

We also consider directed edges to redesign the 15-node controllable networks, which require the minimum control energy. Although a small communication domain ($\gamma = 1.0$) is chosen for each node, the number of edge candidates is increased

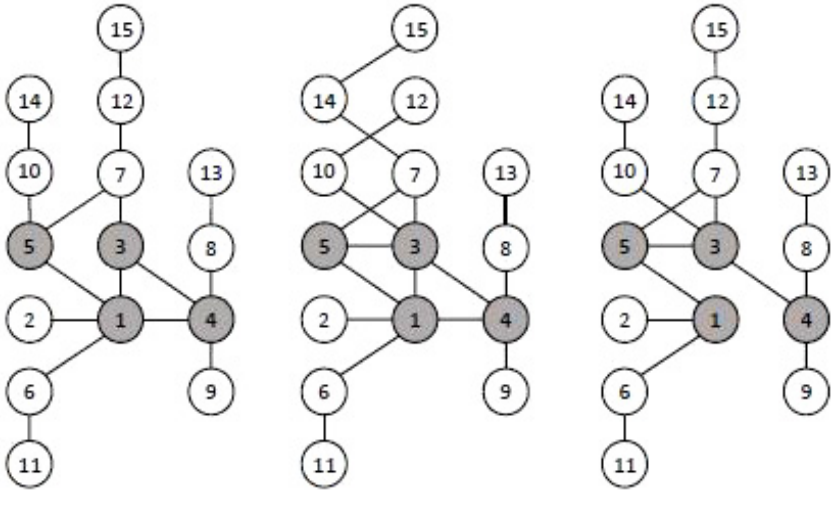


Fig. 10. Edge assignments of undirected 15-node networks with the minimum control energy ($\gamma = 1.5$).

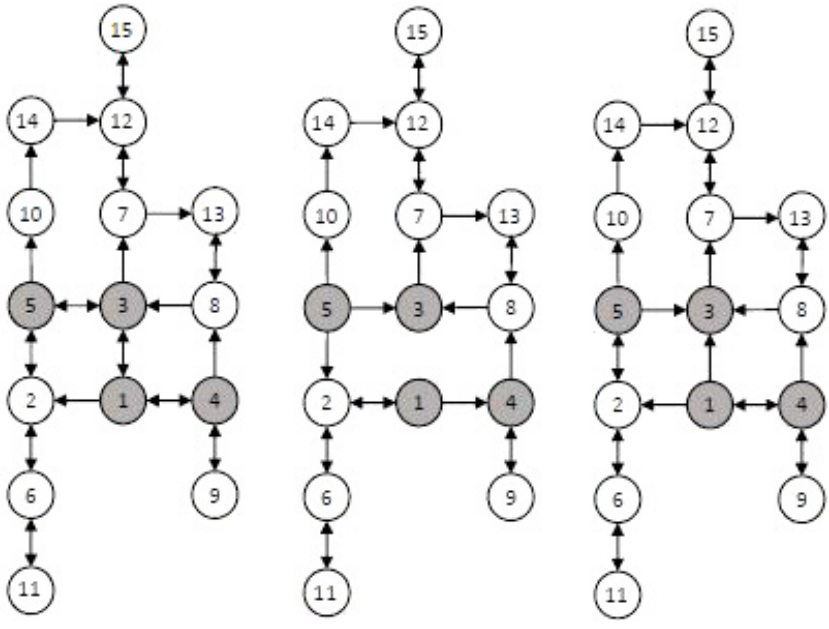


Fig. 11. Edge assignments of unsigned directed 15-node networks with the minimum control energy ($\gamma = 1.0$).

due to the possibility of one-way and two-way communications. Consequently, the network communications are then determined by a 38-bit binary string, $s = \{a_{12}\ a_{21}\ a_{13}\ a_{31}\ \ldots\}$, where a_{ij} represents the communication status (1 for yes or 0 for no) of information passed from node v_j to node v_i. Several design candidates are shown in Fig. 11. A lower control energy (575.11) is achieved by the directed networks, compared to the control energy (679.28) required by the undirected networks.

It should be noted that the discussion above considers unsigned networks. If nodes 4 and 10 have competitive interactions with their neighbors, several signed directed networks with the minimum control energy of 636.56 can be designed, as shown in Fig. 12. The solid lines represent cooperative interactions, while the dashed lines represent competitive interactions.

4.4. *Design of 29-node networks*

We consider 29-node networks, shown in Fig. 13, as another example of applying MAs on leader selections and edge assignments. The network becomes controllable if a minimum of seven leader nodes is properly selected. To design controllable networks with minimum control energy, we consider two different numbers of leader nodes: 7 and 10. Figure 13 includes two final designs. The minimum control energy of the 7-leader network is 6.62×10^4, while the minimum control energy of the 10-leader network is 973.68. The designed networks are undirected. If the 10-leader

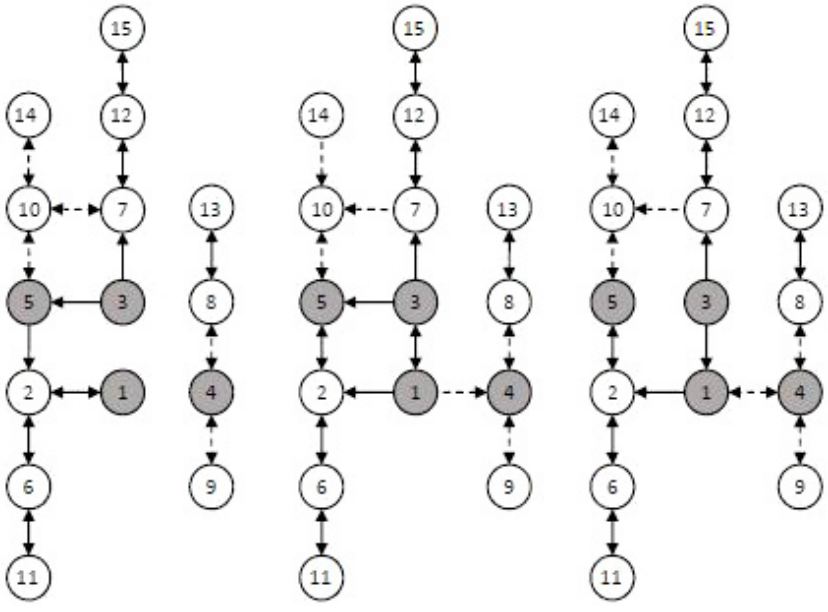
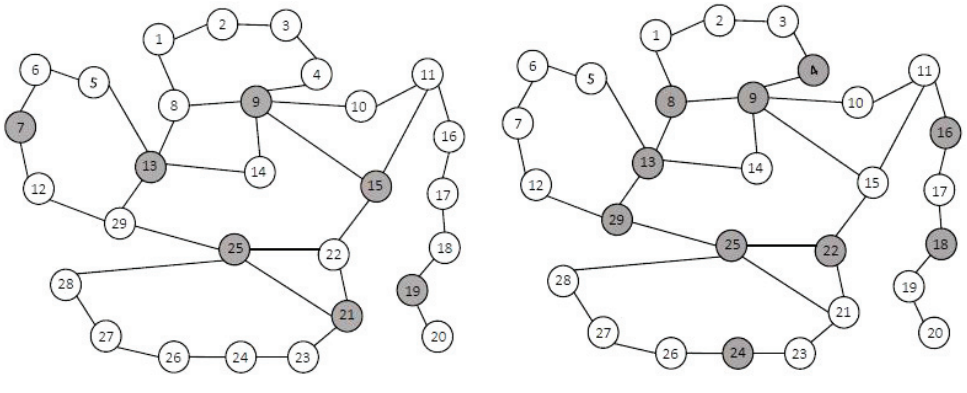


Fig. 12. Edge assignments of signed directed 15-node networks with the minimum control energy ($\gamma = 1.0$).

network is redesigned as a directed network by assigning directed edges as shown in Fig. 14, the control energy can be further reduced to 167.10.

This 29-node network has been employed as an example of leader selection in our previous research [54], in which we investigated the relationship between the network controllability and the graph topology characterization and developed sufficient



(a) (b)

Fig. 13. (a) 7-leader and (b) 10-leader undirected networks with the minimum control energy.

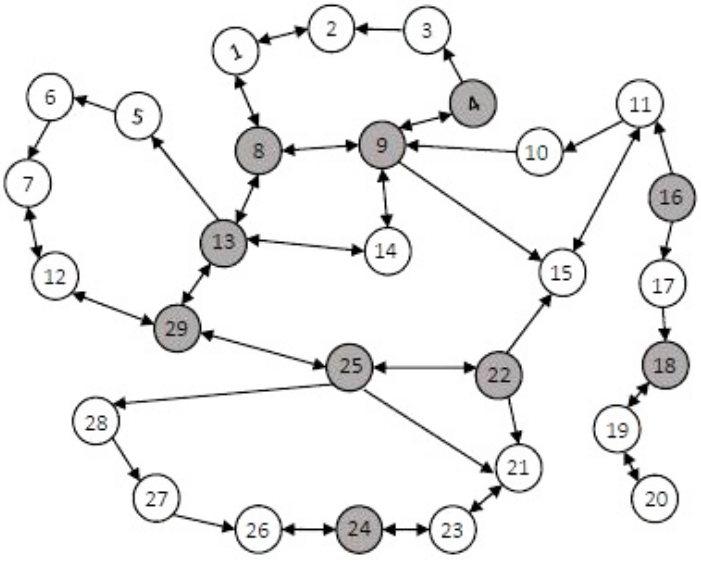


Fig. 14. A 10-leader directed networks with the minimum control energy.

conditions to select leader nodes for network controllability. After initially selecting nodes with degree more than two (i.e. nodes 8, 9, 11, 13, 15, 21, 22, 25, and 29) as the leaders, the graph is partitioned into a set of path and cycle subgraphs, as shown in Fig. 15(a). Then, adding nodes 10 and 14 as additional leaders based on the rules in Theorems 1–4 developed in [54] to ensure the network controllability. Consequently, the designed leader–follower network is controllable and has 11 leader nodes. It would be interesting to compare it with the optimal design (with 11 leaders) in Fig. 15(b), which is obtained via MA simulations. They are two different designs although both are controllable with 11 leader nodes. The previous design in Fig. 15(a) considers the controllability only so that the induced control energy (1.36×10^5) is higher than the control energy of the optimal design, which is 438.84. However, it is observed that the nodes with the highest degrees (4 or 5), including nodes 9, 13, and 25, are selected as leaders in both designs, as well as the other designs in Fig. 13.

A similar phenomenon has been observed in the designs of 15-node networks (see Sec. 4.2), for example, in Fig. 10. According to the graph theory [57], the high-degree nodes can facilitate the partition of the graph into path and cycle subgraphs. If the leaders are selected to ensure the controllability of subgraphs, the whole graph will be controllable by connecting leaders only based on the proved propositions in [54]. Therefore, the high-degree nodes are more likely to be selected as leaders in designing controllable leader–follower networks. However, the aforementioned discussions and conclusions are based on the studies of the graph, including Secs. 4.1–4.4 of this research and our previous research [54]. Those graphs can be easily partitioned as

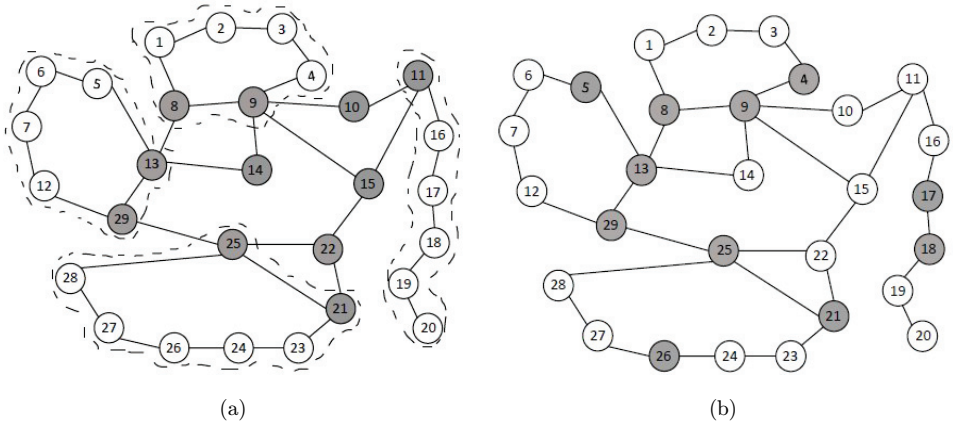


Fig. 15. Two controllable 29-node networks with 11 leaders: (a) The controllability is ensured based on the graph theory [54], and (b) the controllable network with the minimum control energy.

cycle and path subgraphs. To investigate the role of high-degree nodes in leader–follower network design and to demonstrate the application of the proposed method on arbitrary networks, a randomly generated graph is considered in the next section.

4.5. *An arbitrary network*

Here considers an arbitrary unsigned undirected graph consisting of 40 nodes and 124 edges. The degrees of nodes vary between 2 and 12. A minimum of 17 leaders is

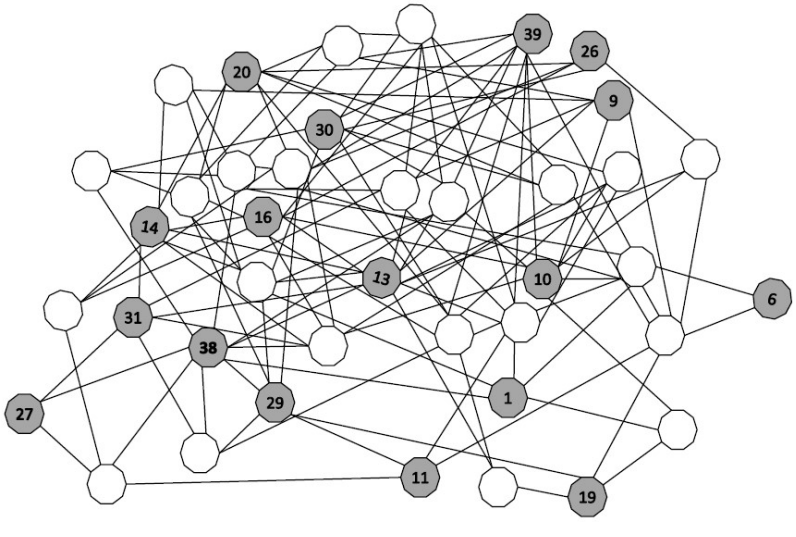


Fig. 16. Seventeen leaders are selected to generate a controllable network with the minimum control energy.

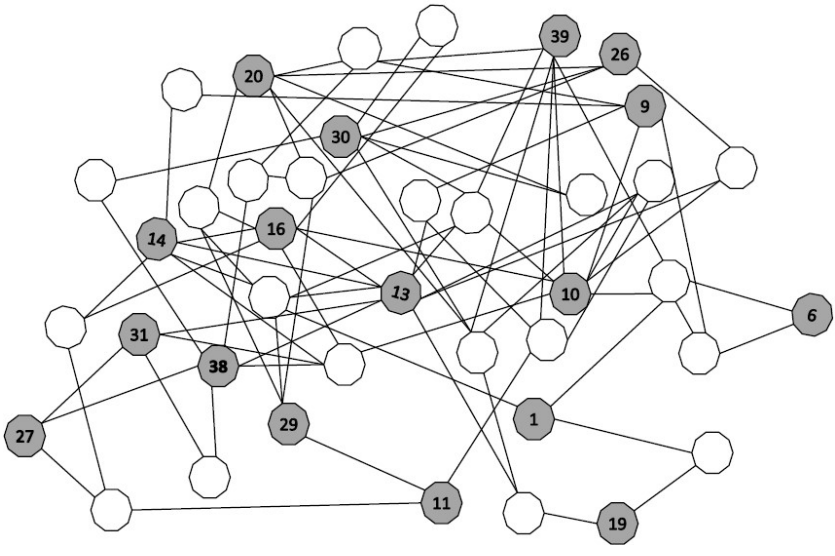
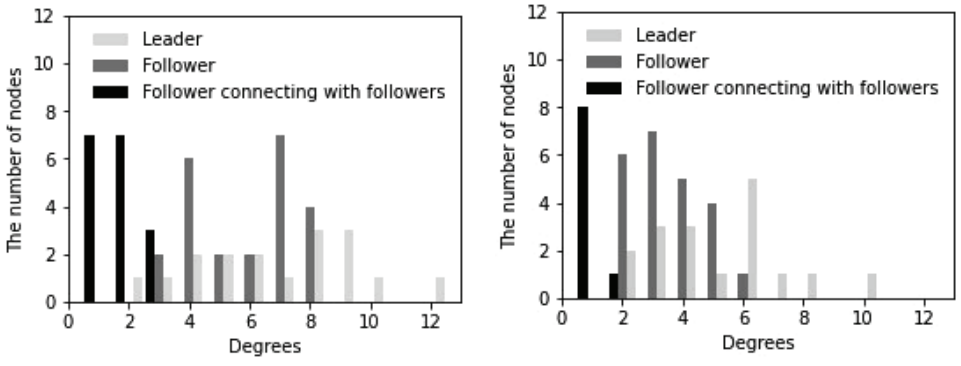


Fig. 17. Edges are reassigned to the 17-node controllable network that has the minimum control energy lower than the original design in Fig. 16.

needed to ensure the controllability of this network. Figure 16 shows one design in which 17 nodes are optimally selected so that the control energy is as minimum as 887.97. We also observed that the nodes with higher degrees are more likely to be selected as leaders. In particular, the top five nodes, including nodes 13, 38, 10, 16, and 39, have the degrees of 12, 10, 9, 9, and 9, respectively. Our simulations showed that they are also selected as leaders in other optimal designs if different numbers of leaders are required.

Since the network is randomly generated, it may have redundant edges from the control aspect. By keeping the same leaders as shown in Fig. 16, we use the existing edges as the potential connections between nodes and redesign the network to achieve the minimum control energy via the edge assignment. Figure 17 illustrates that the new design contains only 82 edges, one-third less than the original network. In addition, leader nodes 13, 38, 10, 16, and 39 keep high degrees that are 10, 6, 8, 7, and 6, respectively. The newly designed network achieves lower control energy that is 561.85, compared to 887.97 as the control energy of the original network in Fig. 16 because redundant connections are eliminated. It shall be noted that removing the leader–leader connection does not affect the control energy [54].

Figure 18 shows the distributions of the number of leaders and the number of followers associated with different degrees. The number of followers that have various follower–follower connections is also illustrated. It can be seen from Fig. 18(a) that the high-degree nodes are more likely to be selected as leaders. Although some followers have average degrees, e.g. up to 8, most of them have only one or two connections with other followers. Since the network is randomly generated, there are



(a) (b)

Fig. 18. Distribution of degrees on the number of nodes in (a) the original design in Fig. 16, and (b) the optimal design in Fig. 17 after edge reassignment.

three nodes each of which has three connections with other followers. However, after edge reassignment, all followers connect with only one or two other followers, shown in Fig. 18(b), although some of them still keep average degrees, i.e. between 4 and 6. In addition, the leaders maintain high degrees.

5. Conclusions and Future Work

In this paper, we presented the first work to design controllable leader–follower networks using MAs. Various optimization problems, including leader selection and edge assignment to pre-defined networks, were considered with the objectives of ensuring controllability and requiring the minimum control energy. The second problem was considered due to the engineering practices in which the number of external control signals is limited in designing a complex system. This problem can be extended to design leader–follower swarms [35] when a limited number of members in the leader group are encapsulated with the desired control signals.

We used 15-node and 29-node networks as examples to demonstrate the application of MAs in network design. The simulation results confirmed that MAs are more efficient and effective than GAs. Specifically, MAs have a faster convergence rate (i.e. less iteration numbers) and are better at finding global optima. Moreover, there are usually multiple optimal network designs, especially for large-scale networks, and MAs have the unique potential to find all of them in one simulation as long as the initial population is large enough and a powerful computing resource is available. Having a full set of optimal designs will help the user make the final decision based on other specifications or constraints that sometimes cannot be easily expressed mathematically. In addition, based on our simulations on simple graphs and arbitrary graphs, the following findings are concluded: (1) high-degree nodes are

more likely to be chosen as the leaders than low-degree nodes; and (2) after the edge assignment to design the controllable network with the minimum control energy, each follower connects with at most two other followers.

In our simulations, a simple hill-climbing was used as the local search operation in MAs. It would be worth examining the use of other local search algorithms, including simulated annealing and tabu search. Also, multi-objective MAs will be considered in future work, for example, considering both the minimum number of leaders (or edges) and the minimum control energy together as the objectives.

The proposed MA framework in this paper can be easily applied to some real-world problems, including power grid design and sensor-actuator placement. However, one challenge relates to the number of potential edges (i.e. communications or interactions) when designing networks via edge assignment. Using the proposed MA framework to solve such problems, the length of the chromosome is assumed to be a constant, which is the same as the number of edge candidates, and every bit in the chromosome directly represents one potential edge. Consequently, if designing a large multi-agent system with mobile robots, the chromosome length would be very long, i.e. $N(N-1)/2$ for undirected networks and $N(N-1)$ for directed networks. To overcome the above challenge, variable chromosomes during MA optimization need to be considered in future work.

It shall be noted that the communication model we employed in this paper is similar to the "sensor coverage model". The role of a communication model is to determine the possible interactions between nodes, which are then identified as communication or noncommunication when designing networks. Other communication models can be adopted, according to the characteristics of networks. For example, the possible communications in a social network (e,g., in a typical office) may depend on the relationships between individuals, and a top-down hierarchical pattern, in which members directly communicate to others at the same level or below but not above, can be applied.

To design large networks containing thousands of nodes, another challenge arises when testing the network controllability. The Kalman criterion is easy to apply if the dimension of the controllability matrix is small. However, it is difficult to numerically determine the rank of the controllability matrix for large real networks because the computing is ill-conditioned and is very sensitive to roundoff errors. Alternatively, the network controllability can be determined via the structural control theory [30] without calculating the rank of the controllability matrix. In that case, the fitness function needs to be revised so that the network controllability can be quantitatively assessed.

In addition, there are various constraints in a real-world complex system, and future work should address how to mathematically represent those constraints in a network model and how to generate the corresponding fitness function when applying MAs to design such networks.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62173314 and Grant U2013601. In addition, Xiao thanks the support from the University of Iowa Technology Institute (ITI) Research Initiative Seed Program.

References

[1] Aguilar, C. O. and Gharesifard, B., A graph-theoretic classification for the controllability of the Laplacian leader-follower dynamics, in *Proc. IEEE Conf. Decision and Control* (Institute of Electrical and Electronics Engineers Inc., Los Angeles, 2014), pp. 619–624, doi: 10.1109/CDC.2014.7039450.

[2] Aguilar, C. O. and Gharesifard, B., Graph controllability classes for the Laplacian leader-follower dynamics, *IEEE Trans. Autom. Control* **60** (2015) 1611–1623.

[3] Bai, W., Li, T. and Tong, S., NN reinforcement learning adaptive control for a class of nonstrict-feedback discrete-time systems, *IEEE Trans. Cybern.* **50** (2020) 4573–4584.

[4] Bai, W., Zhou, Q., Li, T. and Li, H., Adaptive reinforcement learning neural network control for uncertain nonlinear system with input saturation, *IEEE Trans. Cybern.* **50** (2020) 3433–3443.

[5] Balazs, K., Botzheim, J. and Koczy, L., Comparison of various evolutionary and memetic algorithms, in *Integrated Uncertainty Management and Applications*, Huynh, V., Nakamori, Y., Lawry, J. and Inuiguchi, M. (eds.) (Springer, Berlin, 2016), pp. 431–442.

[6] Becker, C. O., Pequito, S., Pappas, G. J. and Preciado, V. M., Network design for controllability metrics, in *2017 IEEE 56th Annual Conf. Decision and Control, CDC 2017*, Vol. 2018 (Institute of Electrical and Electronics Engineers Inc., Melbourne, 2018), pp. 4193–4198, doi: 10.1109/CDC.2017.8264276.

[7] Biesinger, B., Hu, B. and Raidl, G., An evolutionary algorithm for the leader-follower facility location problem with proportional customer behavior, in *Learning and Intelligent Optimization*, Vol. 8426, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) (Springer Verlag, Cham, 2014), pp. 203–217, doi: 10.1007/978-3-319-09584-4_19.

[8] Cassar, I. R., Titus, N. D. and Grill, W. M., An improved genetic algorithm for designing optimal temporal patterns of neural stimulation, *J. Neural Eng.* **14** (2017) 066013.

[9] Chen, C. T., *Linear System Theory and Design*, 4th edn. (Oxford University Press, New York, 2013).

[10] Clark, A., Bushnell, L. and Poovendran, R., On leader selection for performance and controllability in multi-agent systems, in *Proc. IEEE Conf. Decision and Control* (IEEE, Maui, 2012), pp. 86–93, doi: 10.1109/CDC.2012.6426973.

[11] Culberson, J. C., On the futility of blind search: An algorithmic view of "no free lunch", *Evol. Comput.* **6** (1998) 109–127.

[12] Davis, L., *Handbook of Genetic Algorithms* (Van Nostrand Reinhold, New York, 1991).

[13] Estrada, E., Introduction to complex networks: Structure and dynamics, *Lect. Notes Math.* **2126** (2014) 93–131.

[14] Feddema, J. T., Lewis, C. and Schoenwald, D. A., Decentralized control of cooperative robotic vehicles: Theory and application, *IEEE Trans. Rob. Autom.* **18** (2002) 852–864.

[15] Fu, J., Wu, J., Liu, C. and Xu, J., Leaders in communities of real-world networks, *Phys. A* **444** (2016) 428–441.

[16] Goldberg, D. E. and Voessner, S., Optimizing global-local search hybrids, in *Proc. 1st Annual Conf. Genetic and Evolutionary Computation — Vol. 1, GECCO'99* (Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999), pp. 220–228.

[17] Haghighi, R. and Cheah, C. C., Topology-based controllability problem in network systems, *IEEE Trans. Syst. Man Cybern.: Syst.* **47** (2017) 3077–3088.

[18] He, J. and Kang, L., On the convergence rates of genetic algorithms, *Theor. Comput. Sci.* **229** (1999) 23–39.

[19] Hespanha, J. P., *Linear Systems Theory* (Princeton University Press, Princeton, 2018).

[20] Holland, J., *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence* (MIT Press, Cambridge, 1992).

[21] Ishibuchi, H., Yoshida, T. and Murata, T., Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling, *IEEE Trans. Evol. Comput.* **7** (2003) 204–223.

[22] Ji, Z., Lin, H., Cao, S., Qi, Q. and Ma, H., The complexity in complete graphic characterizations of multiagent controllability, *IEEE Trans. Cybern.* **51** (2021) 64–76.

[23] Ji, Z., Lin, H. and Yu, H., Leaders in multi-agent controllability under consensus algorithm and tree topology, *Syst. Control Lett.* **61** (2012) 918–925.

[24] Ji, Z. and Yu, H., A new perspective to graphical characterization of multiagent controllability, *IEEE Trans. Cyber.* **47** (2017) 1471–1483.

[25] Kailath, T., *Linear Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1980).

[26] Konak, A., Coit, D. W. and Smith, A. E., Multi-objective optimization using genetic algorithms: A tutorial, *Reliab. Eng. Syst. Saf.* **91** (2006) 992–1007.

[27] Krasnogor, N. and Smith, J., A tutorial for competent memetic algorithms: Model, taxonomy, and design issues, *IEEE Trans. Evol. Comput.* **9** (2005) 474–488, doi: 10.1109/TEVC.2005.850260.

[28] Leung, K. S., Duan, Q. H., Xu, Z. B., Wong, C. K., Leung, K. S., Wong, C. K., Duan, Q. H. and Xu, Z. B., A new model of simulated evolutionary computation-convergence analysis and specifications, *IEEE Trans. Evol. Comput.* **5** (2001) 3–16.

[29] Li, X. F. and Lu, Z. M., Optimizing the controllability of arbitrary networks with genetic algorithm, *Phys. A* **447** (2016) 422–433.

[30] Liu, Y. Y. and Barabási, A. L., Control principles of complex systems, *Rev. Mod. Phys.* **88** (2016) 035006–035064.

[31] Liu, Y. Y., Slotine, J. J. and Barabási, A. L., Controllability of complex networks, *Nature* **473** (2011) 167–173.

[32] Lu, J., Chen, G., Ogorzalek, M. J. and Trajkovic, L., Theory and applications of complex networks: Advances and challenges, in *Proc. — IEEE Int. Symp. Circuits and Systems* (IEEE, Beijing, 2013), pp. 2291–2294, doi: 10.1109/ISCAS.2013.6572335.

[33] Luca, B. and Craus, M., Local search algorithms for memetic algorithms: Understanding behaviors using biological intelligence, in *2018 22nd Int. Conf. System Theory, Control and Computing, ICSTCC 2018 — Proc.* (Institute of Electrical and Electronics Engineers Inc., Sinaia, 2018), pp. 553–558, doi: 10.1109/ICSTCC.2018.8540690.

[34] Mazurowski, M. A. and Zurada, J. M., Solving decentralized multi-agent control problems with genetic algorithms, in *2007 IEEE Congr. Evolutionary Computation, CEC 2007* (IEEE, Singapore, 2007), pp. 1029–1034, doi: 10.1109/CEC.2007.4424583.

[35] Meng, Z., Lin, Z. and Ren, W., Leader–follower swarm tracking for networked Lagrange systems, *Syst. Control Lett.* **61** (2012) 117–126.

[36] Merz, P., Advanced fitness landscape analysis and the performance of memetic algorithms, *Evol. Comput.* **12** (2004) 303–325.

[37] Merz, P. and Freisleben, B., A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem, in *Proc. 1999 Congr. Evolutionary Computation, CEC 1999*, Vol. 3 (IEEE Computer Society, Washington, DC, 1999), pp. 2063–2070, doi: 10.1109/CEC.1999.785529.

[38] Michalewicz, Z., Janikow, C. Z. and Krawczyk, J. B., A modified genetic algorithm for optimal control problems, *Comput. Math. Appl.* **23** (1992) 83–94.

[39] Mirzaev, I. and Gunawardena, J., Laplacian dynamics on general graphs, *Bull. Math. Biol.* **75** (2013) 2118–2149.

[40] Mousavi, S., Afghah, F., Ashdown, J. D. and Turck, K., Use of a quantum genetic algorithm for coalition formation in large-scale UAV networks, *Ad Hoc Netw.* **87** (2019) 26–36.

[41] Notarstefano, G. and Parlangeli, G., Controllability and observability of grid graphs via reduction and symmetries, *IEEE Trans. Autom. Control* **58** (2013) 1719–1731.

[42] Olshevsky, A., Minimal controllability problems, *IEEE Trans. Control Netw. Syst.* **1** (2014) 249–258.

[43] Olshevsky, A., Eigenvalue clustering, control energy, and logarithmic capacity, *Syst. Control Lett.* **96** (2016) 45–50.

[44] Ottino, J. M., Complex systems, *AIChE J.* **49** (2003) 292–299.

[45] Parlangeli, G. and Notarstefano, G., On the reachability and observability of path and cycle graphs, *IEEE Trans. Autom. Control* **57** (2012) 743–748.

[46] Pasqualetti, F., Zampieri, S. and Bullo, F., Controllability metrics, limitations and algorithms for complex networks, *IEEE Trans. Control Netw. Syst.* **1** (2014) 40–52.

[47] Patterson, S. and Bamieh, B., Leader selection for optimal network coherence, in *Proc. IEEE Conf. Decision and Control* (IEEE, Atlanta, 2010), pp. 2692–2697, doi: 10.1109/CDC.2010.5718151.

[48] Pequito, S., Kar, S. and Aguiar, A. P., Minimum cost input/output design for large-scale linear structural systems, *Automatica* **68** (2016) 384–391.

[49] Pravesjit, S. and Kantawong, K., An improvement of genetic algorithm for optimization problem, in *2nd Joint Int. Conf. Digital Arts, Media and Technology 2017: Digital Economy for Sustainable Growth, ICDAMT 2017* (Institute of Electrical and Electronics Engineers Inc., Chiang Mai, 2017), pp. 226–229, doi: 10.1109/ICDAMT.2017.7904966.

[50] Qu, J., Ji, Z. and Shi, Y., The graphical conditions for controllability of multiagent systems under equitable partition, *IEEE Trans. Cybern.* **51** (2021) 4661–4672.

[51] Rahmani, A., Ji, M., Mesbahi, M. and Egerstedt, M., Controllability of multi-agent systems from a graph-theoretic perspective, *SIAM J. Control Optim.* **48** (2009) 162–186.

[52] Sharapov, R. R. and Lapshin, A. V., Convergence of genetic algorithms, *Pattern Recognit. Image Anal.* **16** (2006) 392–397.

[53] She, B. and Kan, Z., Characterizing controllable subspace and herdability of signed weighted networks via graph partition, *Automatica* **115** (2020) 108900.

[54] She, B., Mehta, S., Ton, C. and Kan, Z., Controllability ensured leader group selection on signed multiagent networks, *IEEE Trans. Cybern.* **50** (2020) 222–232.

[55] She, B., Mehta, S. S., Doucette, E. A., Curtis, J. W. and Kan, Z., Leader group selection for energy-related controllability of signed acyclic graphs, in *Proc. American Control Conf.*, Vol. 2019 (Institute of Electrical and Electronics Engineers Inc., Philadelphia, 2019), pp. 133–138, doi: 10.23919/acc.2019.8815356.

[56] Shimada, Y., Hirata, Y., Ikeguchi, T. and Aihara, K., Graph distance for complex networks, *Sci. Rep.* **6** (2016) 1–6.

[57] Steiner, G., On the k-path partition of graphs, *Theor. Comput. Sci.* **290** (2003) 2147–2155.

[58] Summers, T. H., Cortesi, F. L. and Lygeros, J., On submodularity and controllability in complex dynamical networks, *IEEE Trans. Control Netw. Syst.* **3** (2016) 91–101.

[59] Tang, X., Liu, J. and Zhou, M., Enhancing network robustness against targeted and random attacks using a memetic algorithm, *Europhys. Lett.* **111** (2015) 38005.

[60] Tanner, H. G., On the controllability of nearest neighbor interconnections, in *Proc. IEEE Conf. Decision and Control*, Vol. 3 (IEEE, Nassau, 2004), pp. 2467–2472, doi: 10.1109/cdc.2004.1428782.

[61] Tersi, L., Fantozzi, S. and Stagni, R., Characterization of the performance of memetic algorithms for the automation of bone tracking with fluoroscopy, *IEEE Trans. Evol. Comput.* **19** (2015) 19–30.

[62] Tian, L., Ji, Z., Hou, T. and Yu, H., Bipartite consensus of edge dynamics on coopetition multi-agent systems, *Sci. China Inf. Sci.* **62** (2019) 1–3.

[63] Van De Wal, M. and De Jager, B., Review of methods for input/output selection, *Automatica* **37** (2001) 487–510.

[64] Wicks, M. A. and DeCarlo, R. A., An energy approach to controllability, in *Proc. IEEE Conf. Decision and Control* (IEEE, Austin, 1988), pp. 2072–2077, doi: 10.1109/cdc.1988.194698.

[65] Wrona, S. and Pawełczyk, M., Controllability-oriented placement of actuators for active noise-vibration control of rectangular plates using a memetic algorithm, *Arch. Acoust.* **38** (2013) 529–536.

[66] Wu, J. H., Kalyanam, R. and Givan, R., Stochastic enforced hill-climbing, *J. Artif. Int. Res.* **42** (2011) 815–850.

[67] Xu, X. and He, H. G., A theoretical model and convergence analysis of memetic evolutionary algorithms, in *Advances in Natural Computation*, Vol. 3611, Lecture Notes in Computer Science (Springer, Berlin, 2005), pp. 1035–1043, doi: 10.1007/11539117_142.

[68] Yan, G., Ren, J., Lai, Y. C., Lai, C. H. and Li, B., Controlling complex networks: How much energy is needed? *Phys. Rev. Lett.* **108** (2012) 218703.

[69] Zhao, S. and Pasqualetti, F., Discrete-time dynamical networks with diagonal controllability Gramian, *IFAC-PapersOnLine* **50** (2017) 8297–8302.